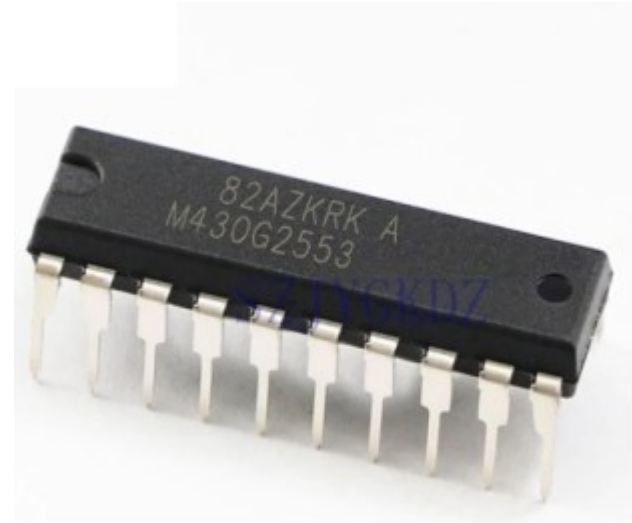


# Token Threading on the MSP430

Brad Rodriguez  
Forth2020 Virtual Meeting  
12 March 2022

# The Target

- MSP430G2553
- 16-bit CPU
- 512 bytes RAM
- 16 KB Flash ROM
  - 8 KB CamelForth kernel
  - 8 KB user application



# OVER + SWAP

- Subroutine Thread

+0		+4		+8	
CALL	adrs of OVER	CALL	adrs of +	CALL	adrs of SWAP

- Address Thread

+0	+2	+4
adrs of OVER	adrs of +	adrs of SWAP

- Token Thread

+0	+1	+2
token OVER	token +	token SWAP

Adrs of +
Adrs of DROP
Adrs of DUP
Adrs of OVER
Adrs of SWAP



# Extensible vs. Hybrid

- All words have tokens
  - Extensible token table
  - Large or “creative” encoding
- Selected subset has tokens
  - Fixed size token table
  - Fixed size token (e.g. 8 bits)
  - Mechanism to reference other words

# Hybrid Token/Non-Token Model

- CALL token

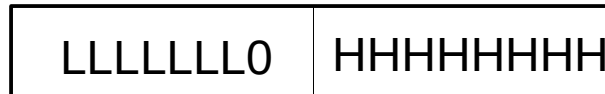


- Special encoding

- Token



- Address



# ITC vs TTC NEXT

- ITC

```
.macro NEXT
    MOV @IP+,W
    MOV @W+,PC
.endm
```

## TTC

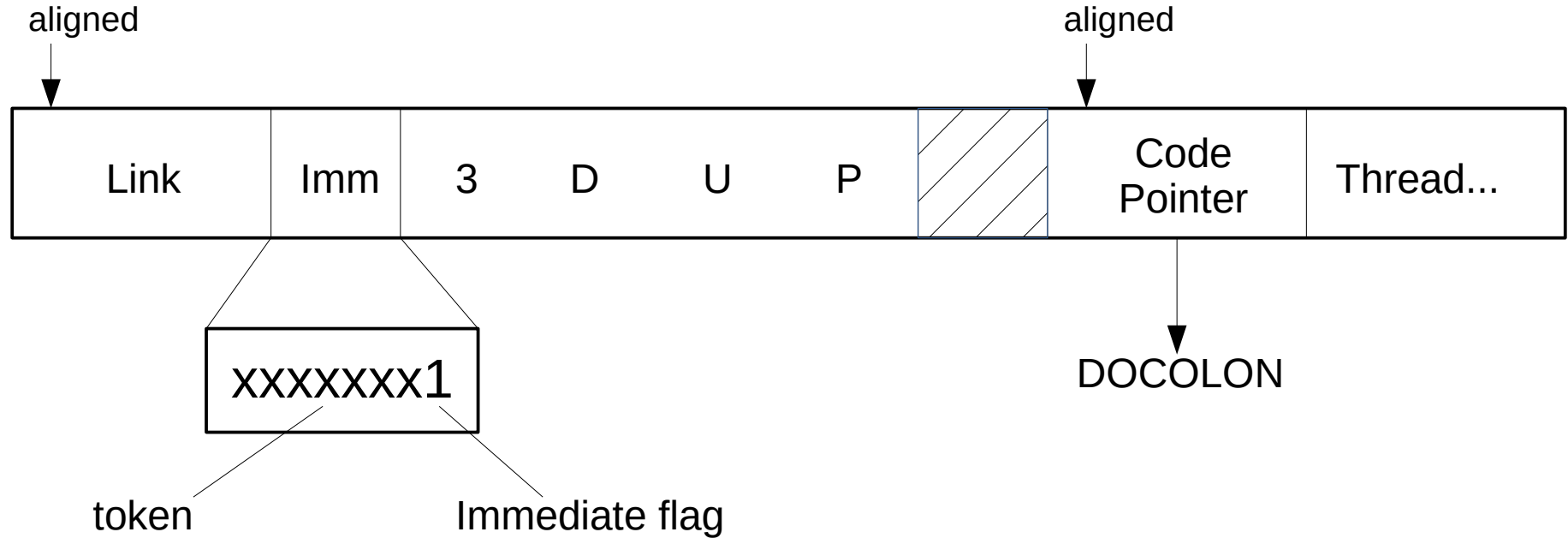
```
_NEXT_: MOV.B @IP+,W
        BIT.B #1,W
        JNZ _ISTOKEN_
; not a token - fetch address
        MOV.B @IP+,X
        SWPB X
        BIS X,W
        MOV @W+,PC

_ISTOKEN_: ; W contains 00tt
        MOV TOKENTABLE-1(W),W
        MOV @W+,PC
```

# The Token Table

	<b>x1</b>	<b>x3</b>	<b>x5</b>	<b>x7</b>	<b>x9</b>	<b>xB</b>	<b>xD</b>	<b>xF</b>
<b>0x</b>	0	1	2	-1	1CHARS	CELL	lit	clit
<b>1x</b>	EXIT	DUP	?DUP	DROP	SWAP	OVER	ROT	NIP
<b>2x</b>	TUCK	>R	R>	R@	SP@	SP!	RP@	RP!
<b>3x</b>	@	!	C@	C!	I@	I!	IC@	IC!
<b>4x</b>	+	+!	M+	-	AND	OR	XOR	INVERT
<b>5x</b>	NEGATE	><	1+	1-	2*	2/	LSHIFT	RSHIFT
<b>6x</b>	0=	0<	=	<>	<	>	U<	U>
<b>7x</b>	branch	?branch	(do)	(loop)	(+loop)	UNLOOP	I	J
<b>8x</b>	EXECUTE	UM*	UM/MOD	EMIT	KEY	KEY?	CELL+	CHAR+
<b>9x</b>	FILL	CMOVE	CMOVE>	D->I	SKIP	SCAN	S=	N=
<b>Ax</b>	*	/	MOD	*/	/MOD	*/MOD	MAX	MIN
<b>Bx</b>	UMAX	UMIN	2DUP	2DROP	2@	2!	2SWAP	2OVER
<b>Cx</b>	TYPE	SPACE	CR	<#	#	#S	HOLD	#>
<b>Dx</b>	LATEST	IDP	IHERE	IALLLOT	HERE	ALLOT	,XT	I,
<b>Ex</b>	IC,	CCLRB	CSETB	CTSTB	CLRB	SETB	TSTB	CELLS
<b>Fx</b>	sbranch	s?branch						NOOP

# Forth Word Layout





# What should the 'xt' be?

- 'xt' is
  - if odd, an 8-bit token
  - if even, a 16-bit CFA
- Produces xt: FIND ' ['
- Consumes xt: EXECUTE COMPILE, >BODY  
HEX ' DUP U. 13 ok

# Other Optimizations

- Short literals (-128..+127)
- Short branch offsets
  - branch and ?branch only

# Results

- ITC MSP430 CamelForth 0.5
  - 7276 bytes ( < 7.5 KiB )
  - DO I DROP LOOP takes 3.88 usec
- TTC MSP430 CamelForth 0.1
  - 6580 bytes ( < 6.5 KiB )
  - DO I DROP LOOP takes 7.50 usec

# Future Work

- Token Action, Token Thread



A table with five rows, representing the memory layout for the token thread. An arrow from the 'Code Token' field in the diagram above points to the first row. The rows contain the following text:

Adrs of DOCOLON
Adrs of DOVAR
Adrs of DOCON
...
Adrs of DOCODE

# Questions?

-